

DNA Prefix

Big-O Blue Lecture 14: Trie

Tóm tắt đề bài

Tóm tắt đề bài

- Cho N mẫu DNA, mỗi mẫu là một xâu kí tự chỉ gồm các chữ cái {"A", "C", "G", "T"}.
- Chọn ra một tập con các mẫu DNA sao cho: $\langle \text{Độ dài tiền tố chung dài nhất} \rangle \times \langle \text{Số lượng mẫu được chọn} \rangle$ là lớn nhất.
- In ra tích lớn nhất tìm được.

Mô tả Input/Output

Input

Dòng đầu tiên chứa số lượng bộ test T ($T \leq 10$), mỗi bộ test gồm có:

- Số lượng chuỗi n ($1 \leq n \leq 50.000$)
- n dòng tiếp theo mỗi dòng chứa một chuỗi.

Output

In ra “Case x : ans” với x là số thứ tự của bộ test, ans là kết quả lớn nhất nhận được khi thực hiện phép tính.

Giải thích ví dụ

Ví dụ 1

A C G T

A C G T G C G T

A C C G T G C

A C G C C G T

9

Ví dụ 2

C G C G C G C G C G C G C C C C G C C C G C G C

C G C G C G C G C G C G C C C C G C C C G C A C

66

C G C G C G C G C G C G C C C C G C C C G C T C

Ví dụ 3

C G C G C C G C G C G C G C G C G C G C

G G C G C C G C G C G C G C G C G C T C

20

Hướng dẫn giải

Nhận xét

Duyệt qua tất cả các tiền tố, với mỗi tiền tố:

- + Đếm xem có bao nhiêu xâu có tiền tố này, gọi là *count*.
- + Tính $\langle \text{độ dài tiền tố} \rangle \times \text{count}$ và so sánh với kết quả tối ưu hiện tại.



Sử dụng cây Trie, mỗi node của cây:

- + Tương ứng với một tiền tố, độ dài tiền tố là khoảng cách từ node đó tới root.
- + Có thuộc tính *count* là số lượng xâu có tiền tố tương ứng.

Các bước giải

Bước 1: Chuẩn bị:

- + Node gốc của cây trie: *root*
- + Kết quả: *ans* = 0

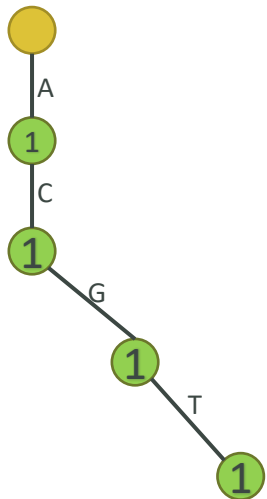
Bước 2: Lần lượt đọc và thêm các xâu vào cây Trie:

- Gán *curNode* = *root*
- Duyệt qua từng ký tự *c* của xâu:
 - + Nếu node con của *curNode* ứng với *c* chưa tồn tại → tạo mới
 - + Tăng *count* của node con này lên 1 đơn vị.
 - + Cập nhật *ans* với: $\langle count \text{ của node con} \rangle \times \langle \text{độ dài tiền tố} \rangle$.
 - + Gán lại *curNode* bằng node con.

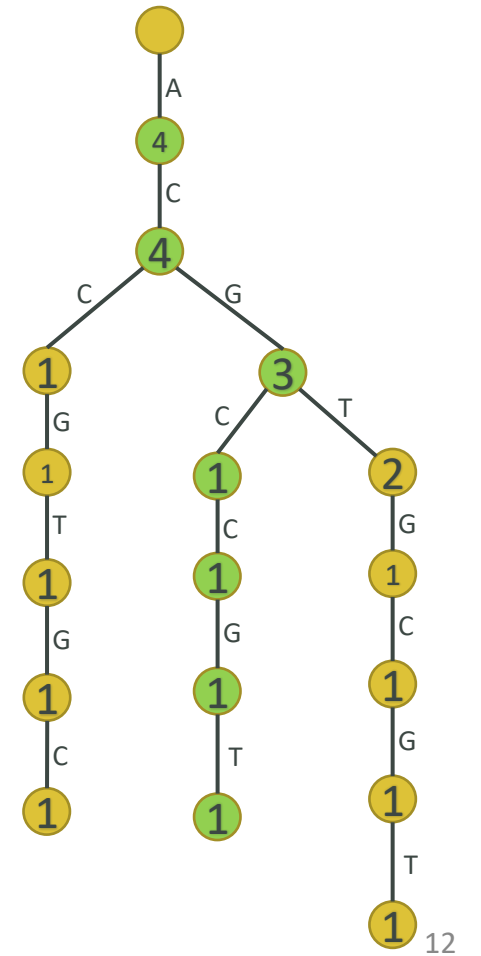
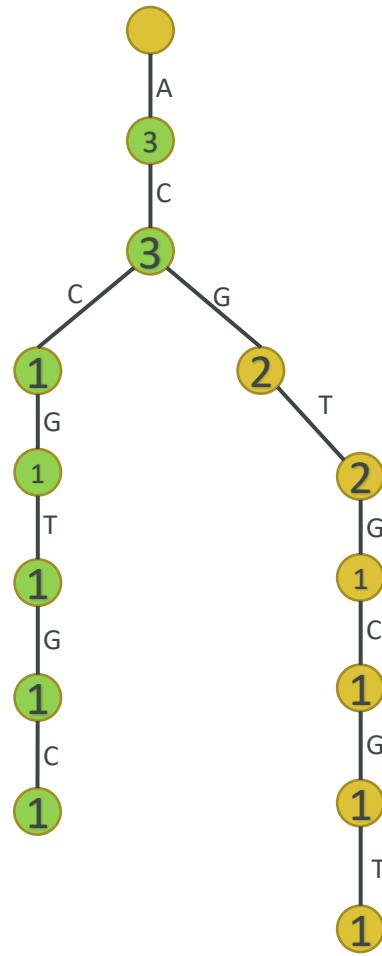
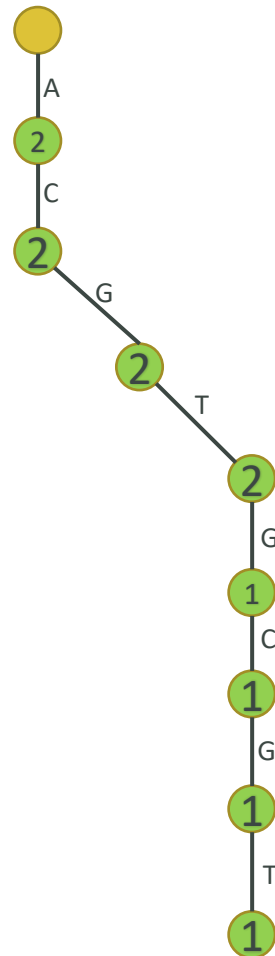
Bước 3: Xuất *ans* ra màn hình

Độ phức tạp: $O(T \cdot N \cdot \text{length}(\text{DNA}))$

Minh họa thuật toán



4
ACGT
ACGTGCGT
ACCGTGC
ACGCCGT



Mã giả

Mã giả

```
class Node():
    count = 0
    child[0..3] = {null..null}

function addWord(root,s):
    curNode = root
    for i = 0 -> len(s) - 1:
        idx = index[s[i]]
        if curNode.child[idx] == null:
            curNode.child[idx] = Node()
        curNode = curNode.child[idx]
        curNode.count += 1
    ans = max(ans,curNode.count*(i+1))
```

```
read(t)
for i = 0 -> t
    read(n)
    root = Node()
    index = {"A":0, "C":1, "G":2, "T":3}
    ans = 0
    for i = 1 -> n:
        read(s)
        addWord(root,s)
    print(ans)
```

Thank you